CoreJava:

1.1

**package** Core;

**public** **enum** RoomType {

***GENERAL***(2000),***SPECIAL***(5000),***SEMI***(3000),***ICU***(70000);

**private** **double** roomCost;

**private** RoomType(**double** roomCost) {

**this**.roomCost = roomCost;

}

**public** **double** getRoomCost() {

**return** roomCost;

}

}

**package** Core;

**import** java.time.LocalDate;

**public** **class** Patient {

**private** **int** serialNo;

**private** String patientname;

**private** **int** age;

**private** String gender;

**private** String address;

**private** String phoneNumber;

**private** String disease;

**private** LocalDate admitDate;

**private** **double** annualIncome;

**private** RoomType roomType;

**private** **static** **int** *counterId*;

**static**

{

*counterId*=100;

}

**public** Patient(String patientname, **int** age, String gender, String address, String phoneNumber, String disease,

LocalDate admitDate, **double** annualIncome, RoomType roomType) {

**super**();

**this**.serialNo=*counterId*++;

**this**.patientname = patientname;

**this**.age = age;

**this**.gender = gender;

**this**.address = address;

**this**.phoneNumber = phoneNumber;

**this**.disease = disease;

**this**.admitDate = admitDate;

**this**.annualIncome = annualIncome;

**this**.roomType = roomType;

}

@Override

**public** String toString() {

**return** "Patient [serialNo=" + serialNo + ", patientname=" + patientname + ", age=" + age + ", gender=" + gender

+ ", address=" + address + ", phoneNumber=" + phoneNumber + ", disease=" + disease + ", admitDate="

+ admitDate + ", annualIncome=" + annualIncome + ", roomType=" + roomType + "]";

}

**public** RoomType getRoomType() {

**return** roomType;

}

**public** String getGender() {

**return** gender;

}

**public** String getDisease() {

**return** disease;

}

**public** **void** setRoomType(RoomType roomType) {

**this**.roomType = roomType;

}

}

1.2

**package** CustomException;

**public** **class** PatientExceptionHandling **extends** Exception{

**public** PatientExceptionHandling(String msg)

{

**super**(msg);

}

}

**package** Validation;

**import** java.time.LocalDate;

**import** Core.Patient;

**import** Core.RoomType;

**import** CustomException.PatientExceptionHandling;

**public** **class** ValidationRules {

**public** **static** Patient validateAllInputs(String patientname, **int** age, String gender, String address, String phoneNumber, String disease,

String admitDate, **double** annualIncome, String roomType) **throws** PatientExceptionHandling

{

LocalDate validateDate = *parseAndValidateDate*(admitDate);

RoomType validateRoom = *parseAndValidateRoom*(roomType);

**return** **new** Patient(patientname, age, gender,address,phoneNumber,disease,validateDate,annualIncome,validateRoom);

}

**public** **static** RoomType parseAndValidateRoom(String room)

{

return RoomType.*valueOf*(room.toUpperCase());

}

**public** **static** LocalDate parseAndValidateDate(String date) **throws** PatientExceptionHandling

{

LocalDate admitDate=LocalDate.*parse*(date);

LocalDate today=LocalDate.*now*();

**if**(admitDate.isEqual(today))

**return** admitDate;

**throw** **new** PatientExceptionHandling("Please Enter the correct Date");

}

}

1.3

**package** Utils;

**import** java.io.FileInputStream;

**import** java.io.FileNotFoundException;

**import** java.io.FileOutputStream;

**import** java.io.IOException;

**import** java.io.ObjectInputStream;

**import** java.io.ObjectOutputStream;

**import** java.util.ArrayList;

**import** java.util.List;

**import** Core.Patient;

**public** **class** IOUtils {

**public** **static** **void** storePatientDetails(String fileName,List<Patient>patient) **throws** IOException

{

**try**(ObjectOutputStream out=**new** ObjectOutputStream(**new** FileOutputStream(fileName)))

{

out.writeObject(patient);

}

}

// @SuppressWarnings("unchecked")

// public static List<Patient> restorePatientDetails(String fileName)

// {

// try(ObjectInputStream in=new ObjectInputStream(new FileInputStream(fileName)))

//

// {

// return (List<Patient>) in.readObject();

// }catch(Exception e)

//

// {

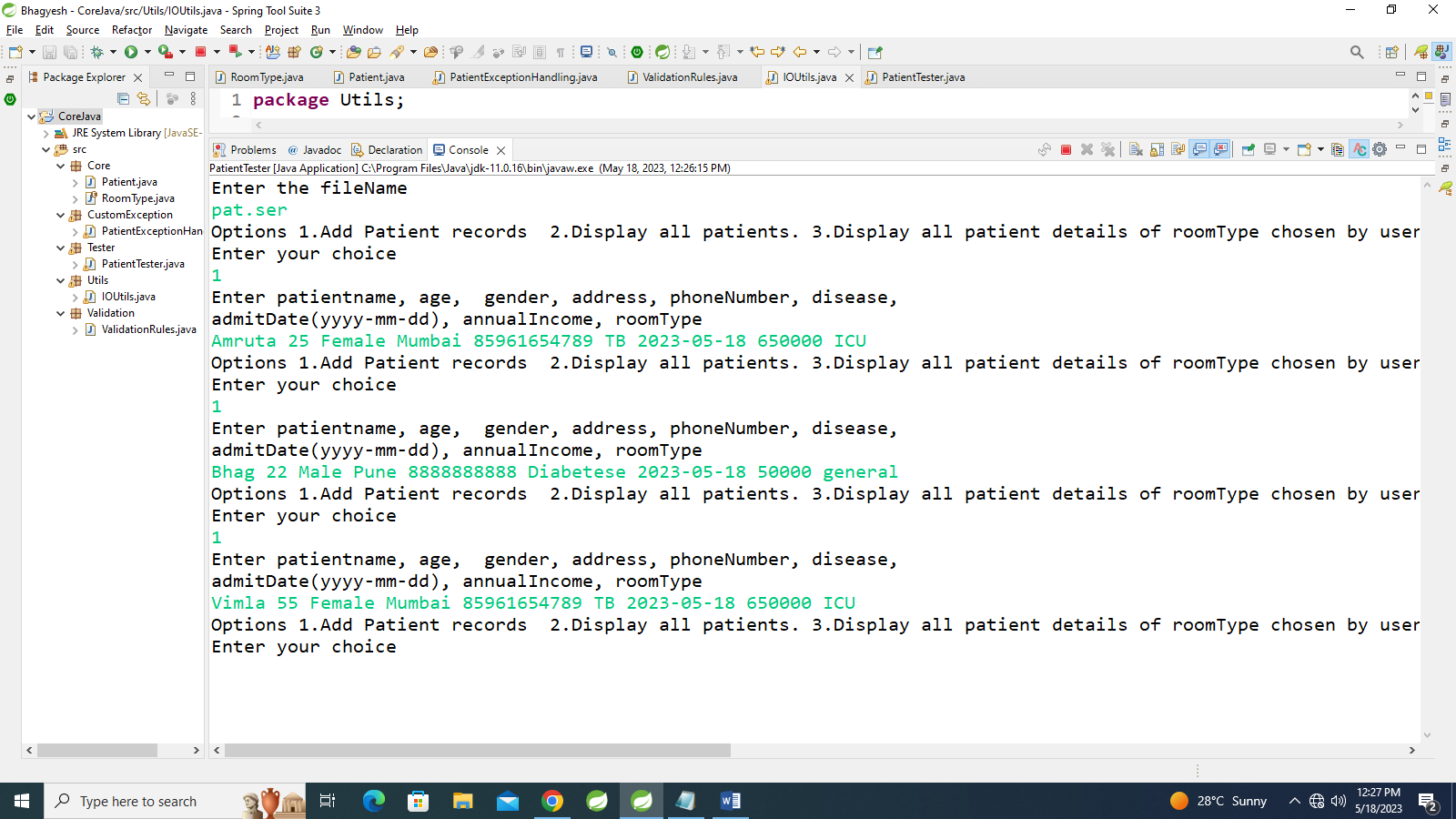
// return ArrayList<Patient>pat=new ArrayList<Patient>();

// }

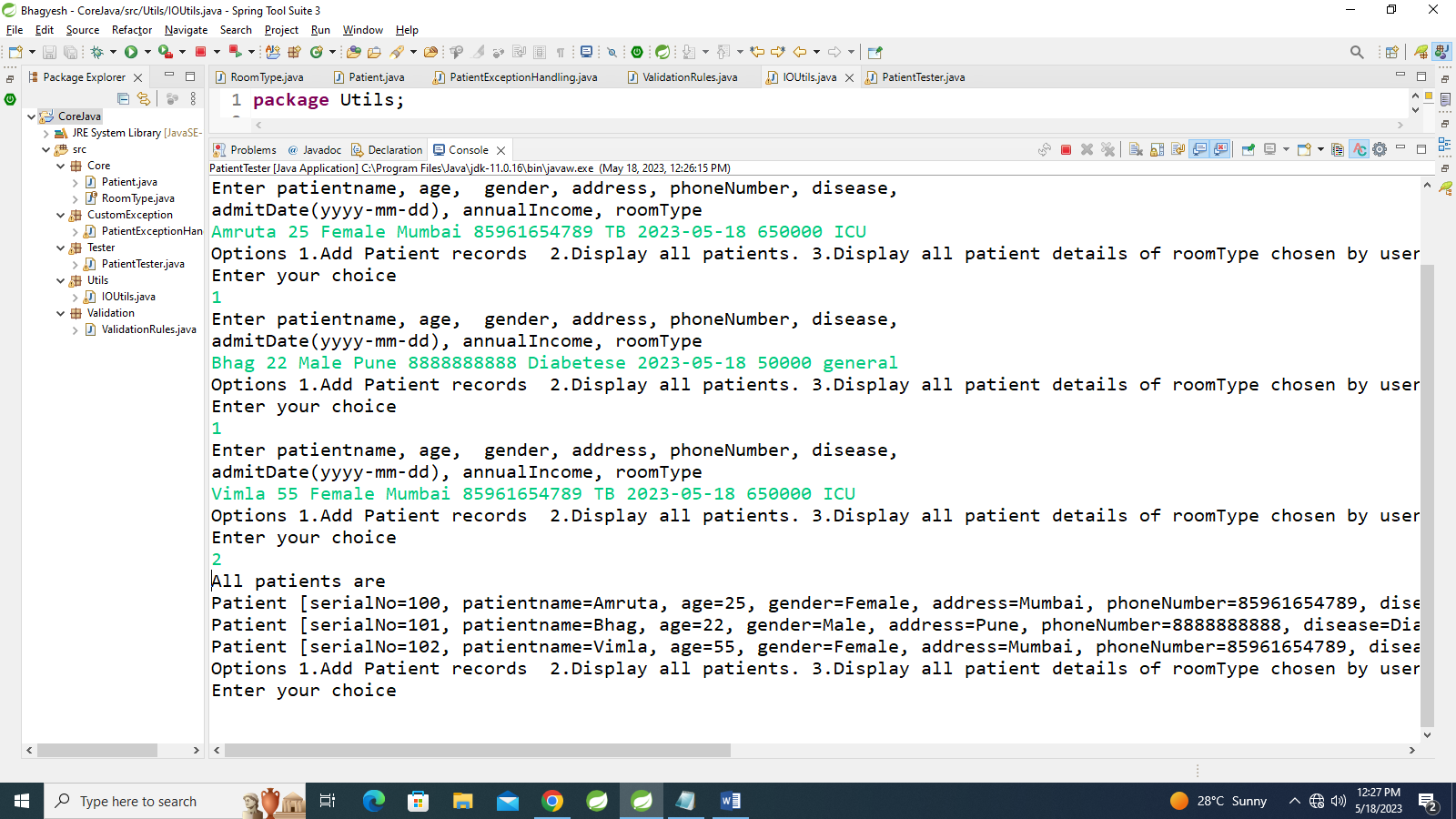
// }

}

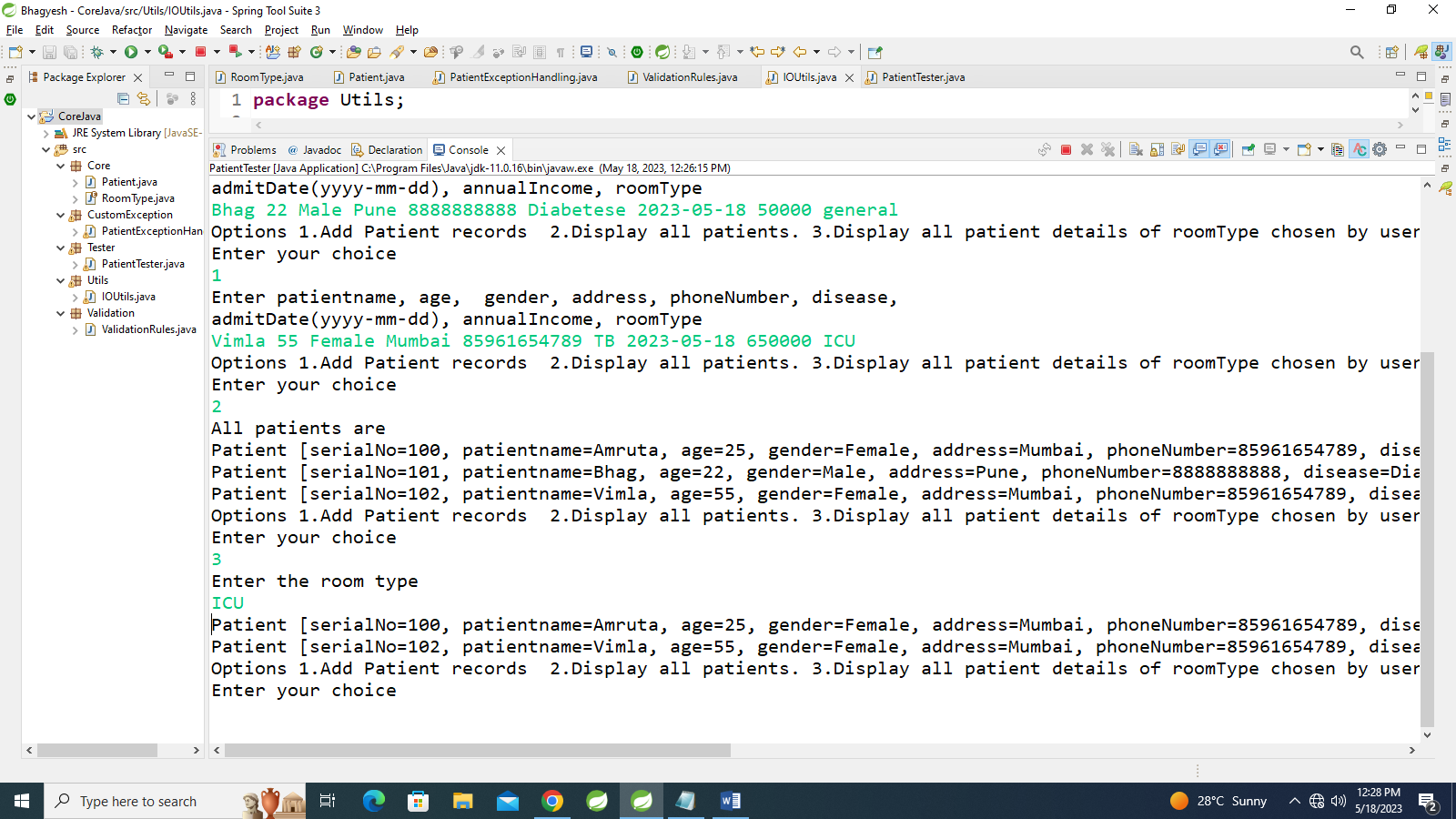
2.1)

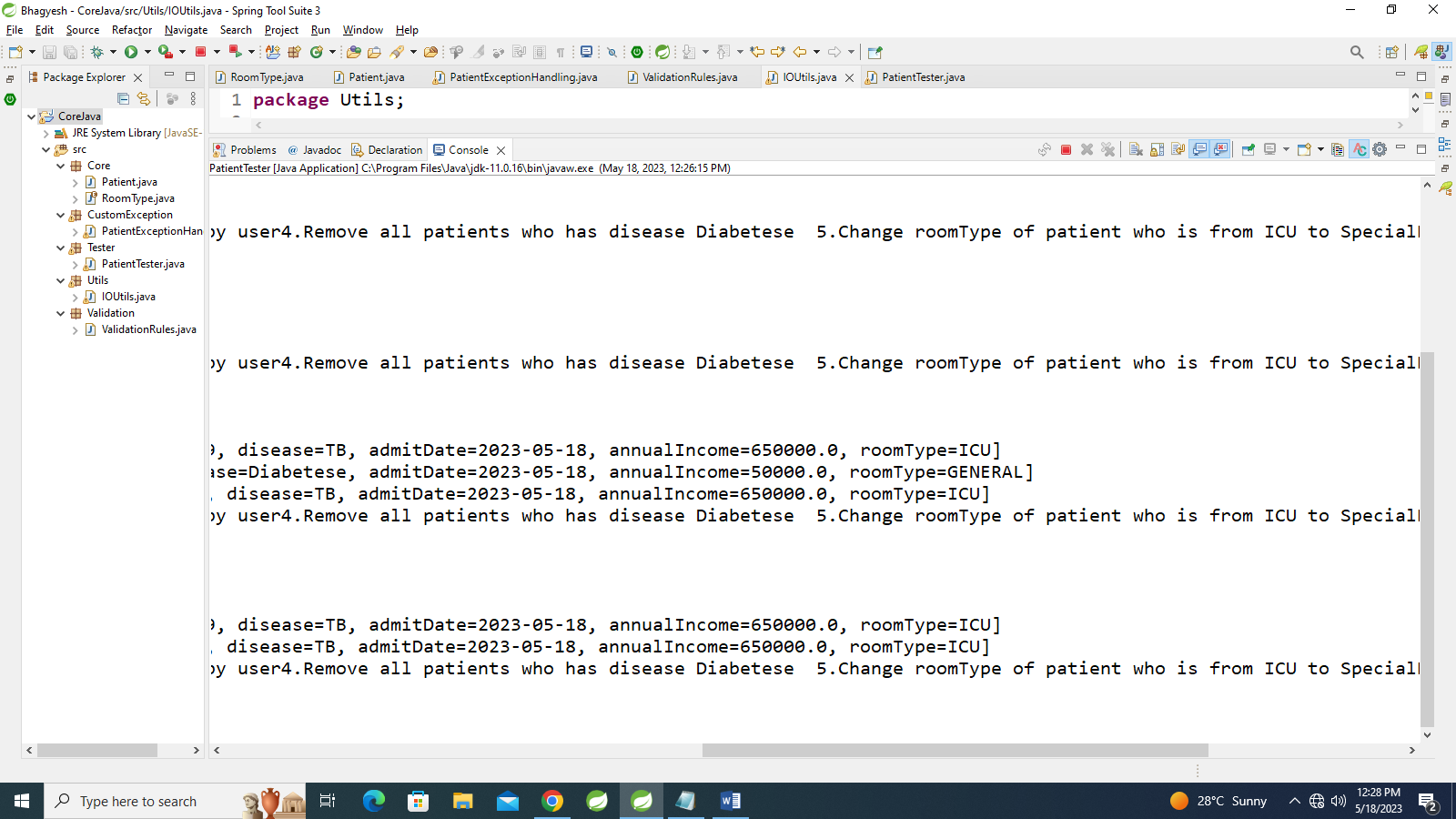


2.2)



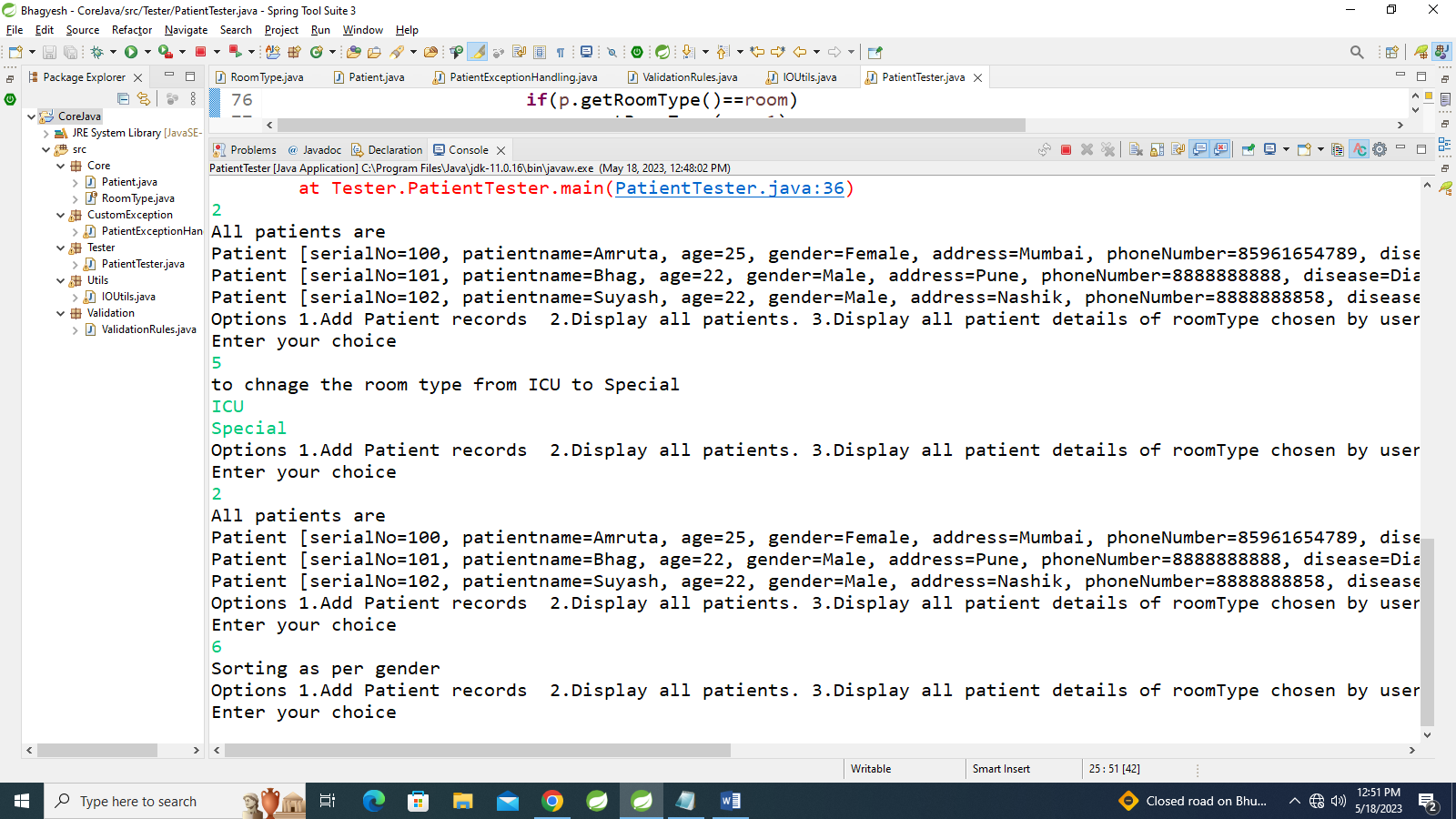
2.3)



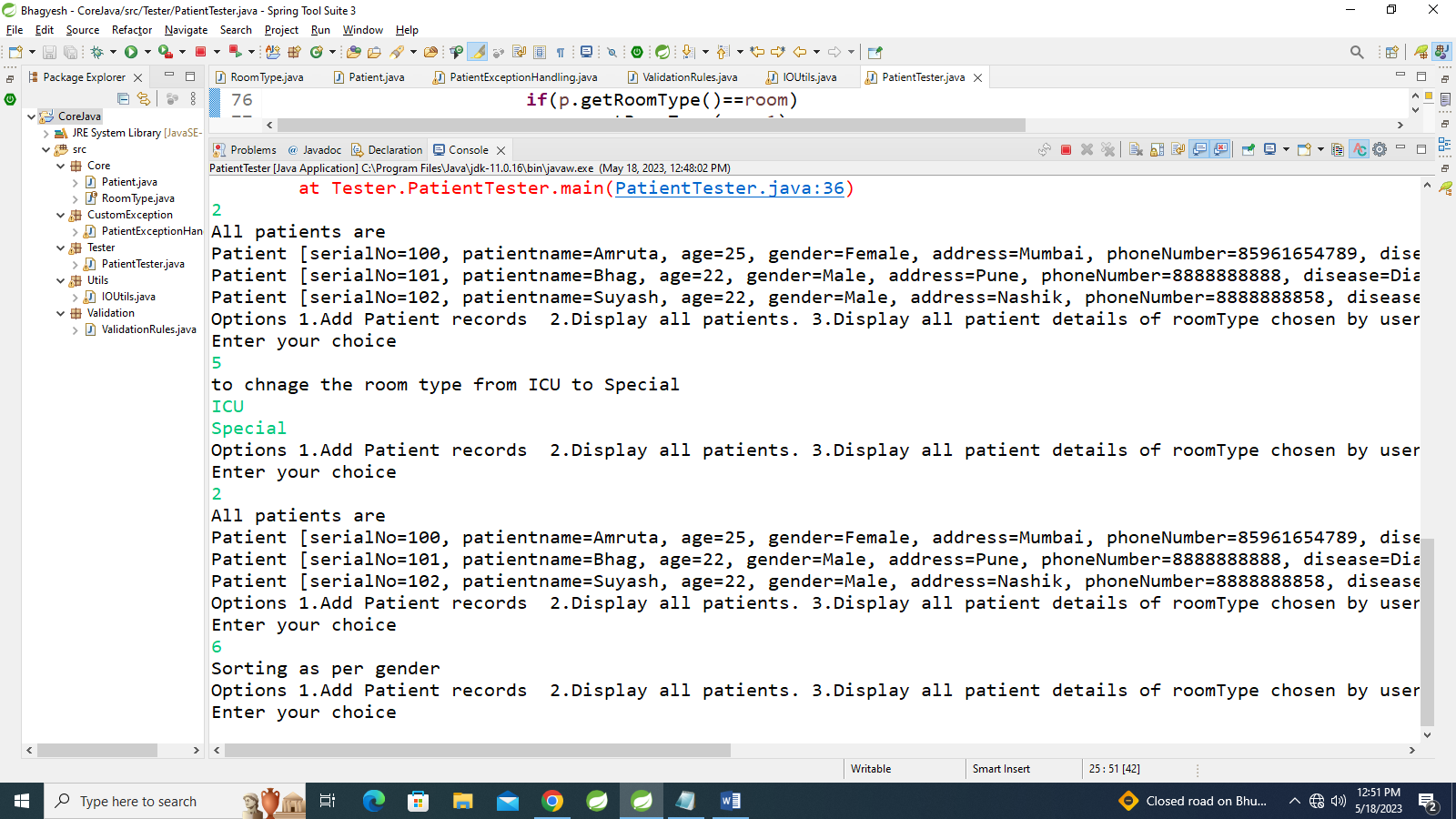


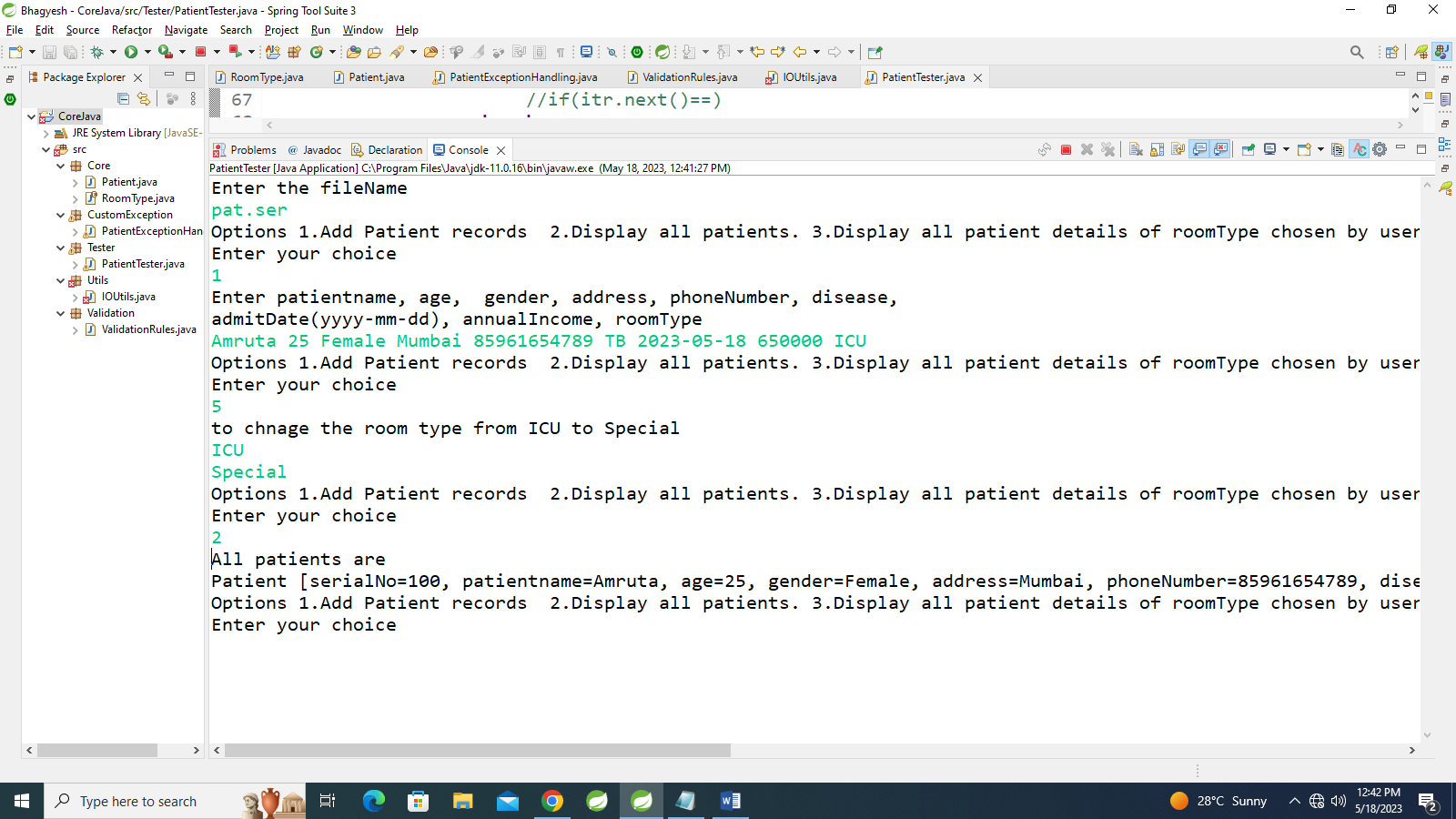
2.4)

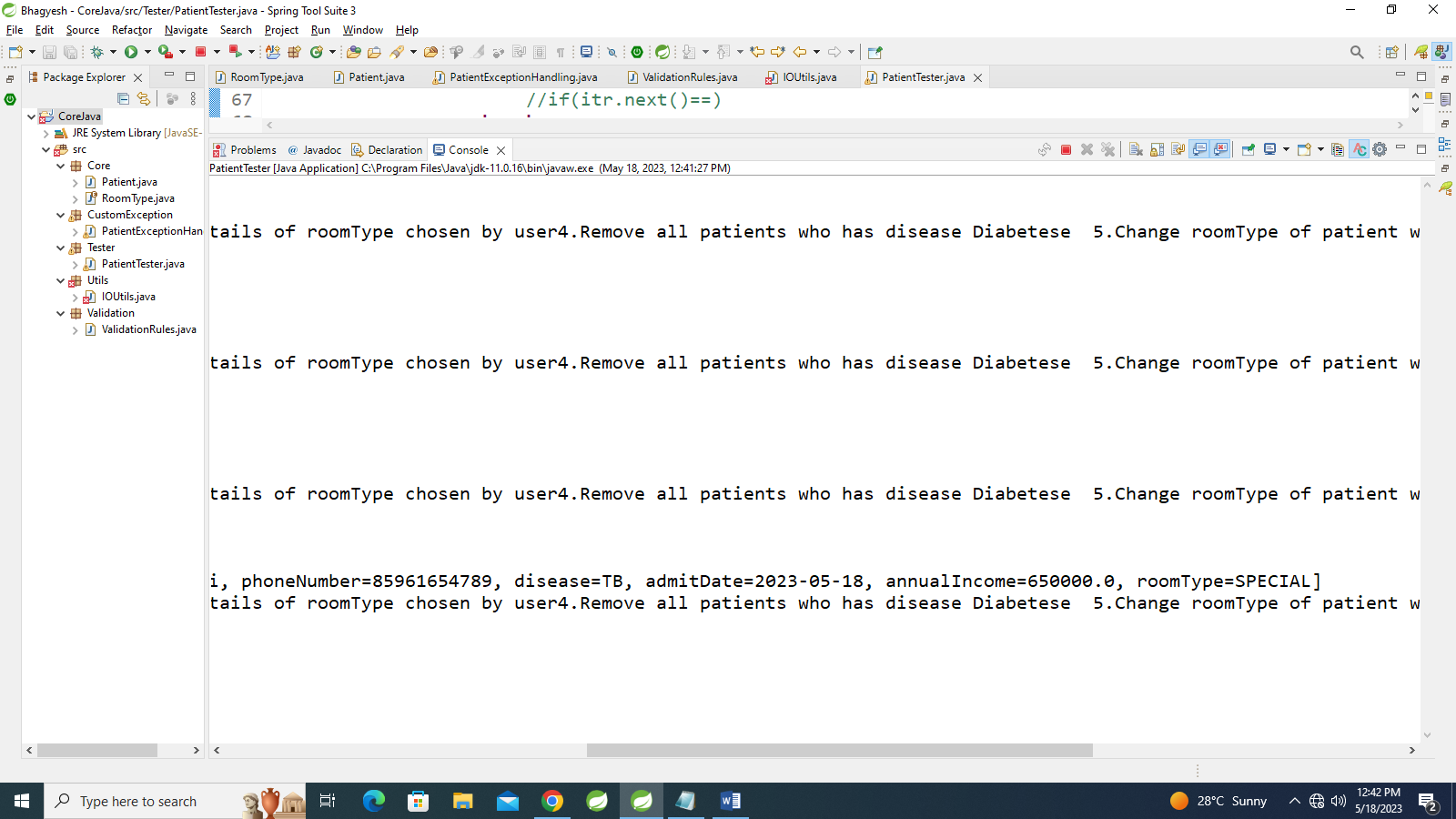
2.5)



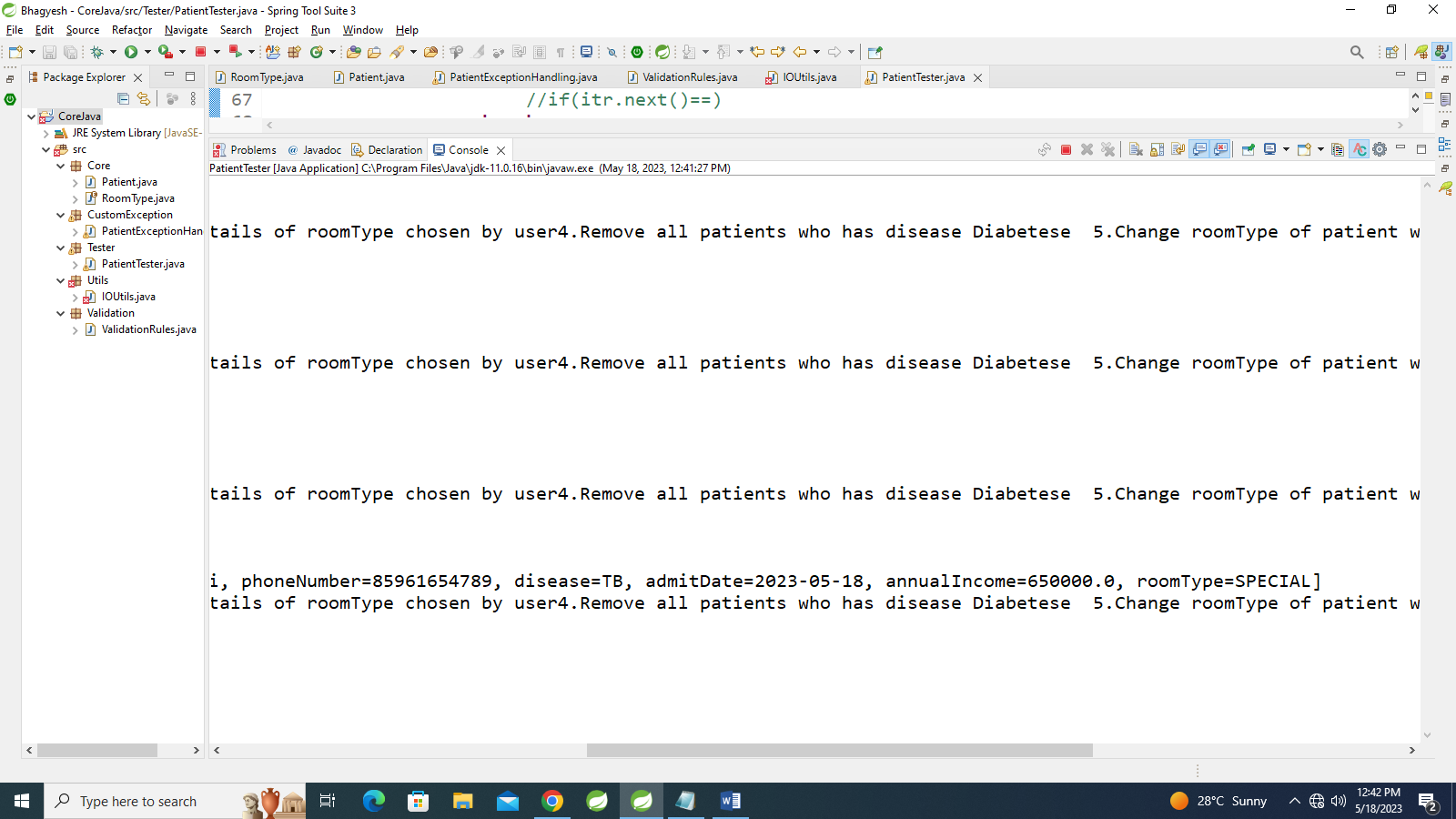
2.6)

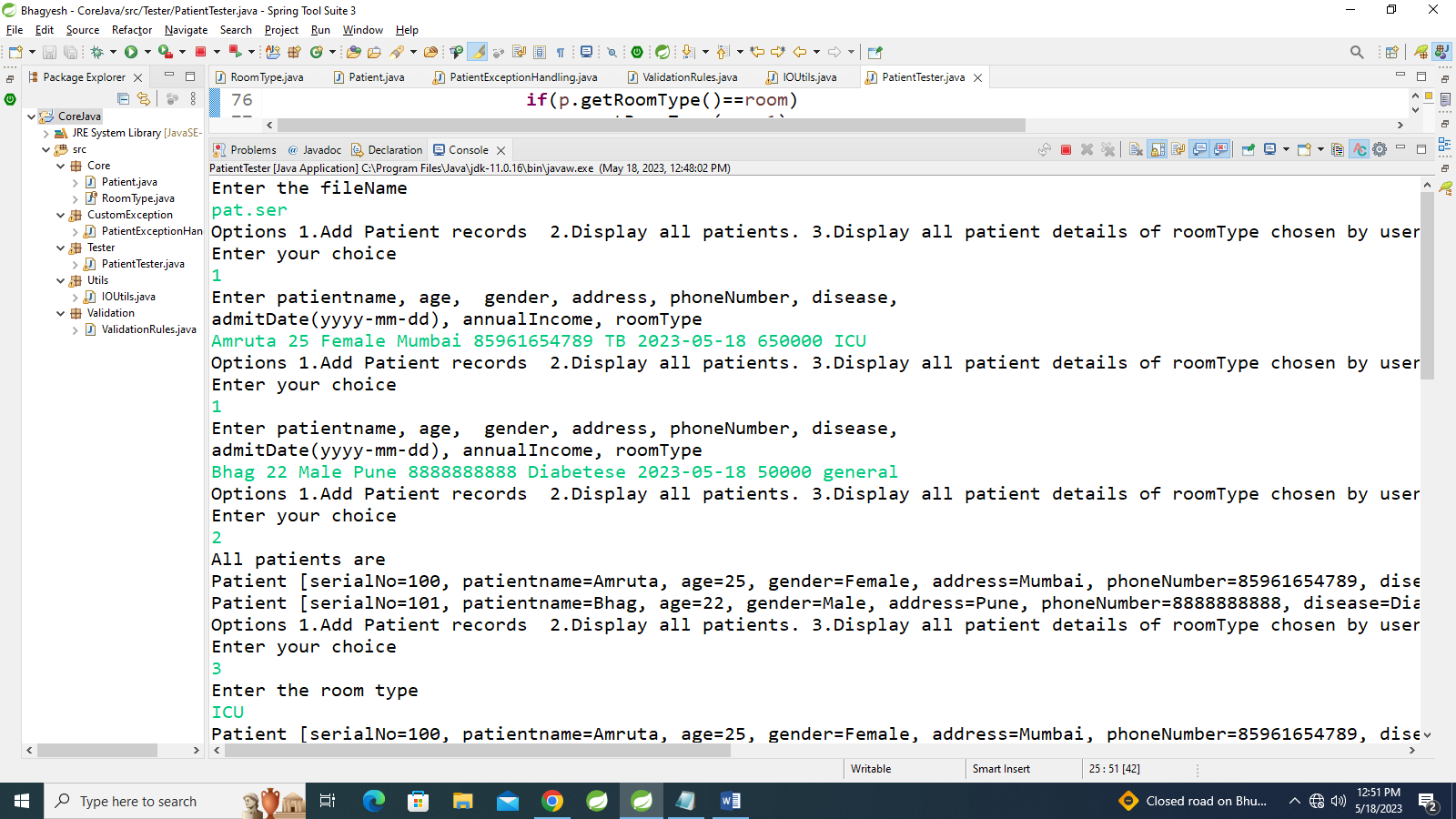


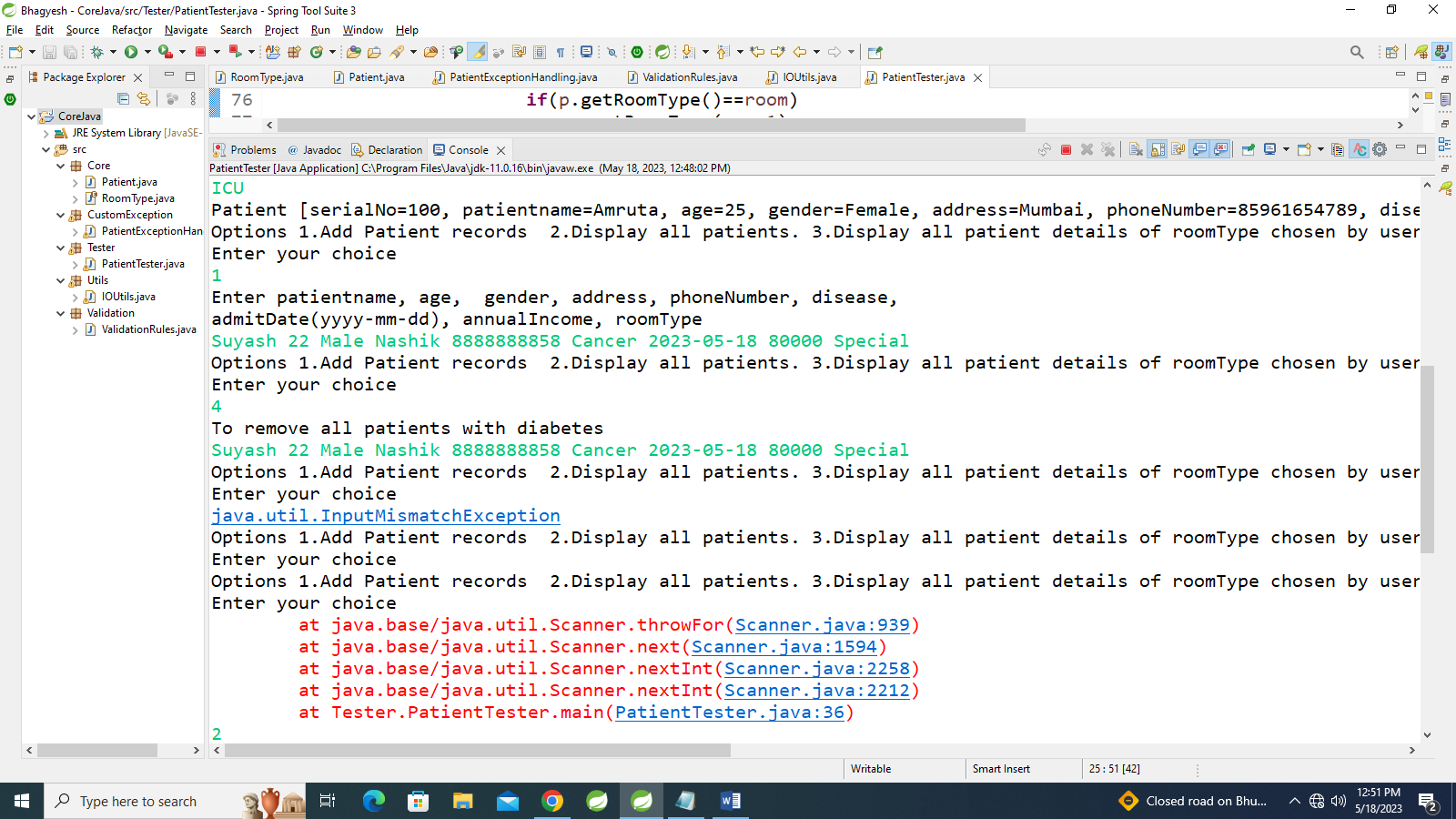


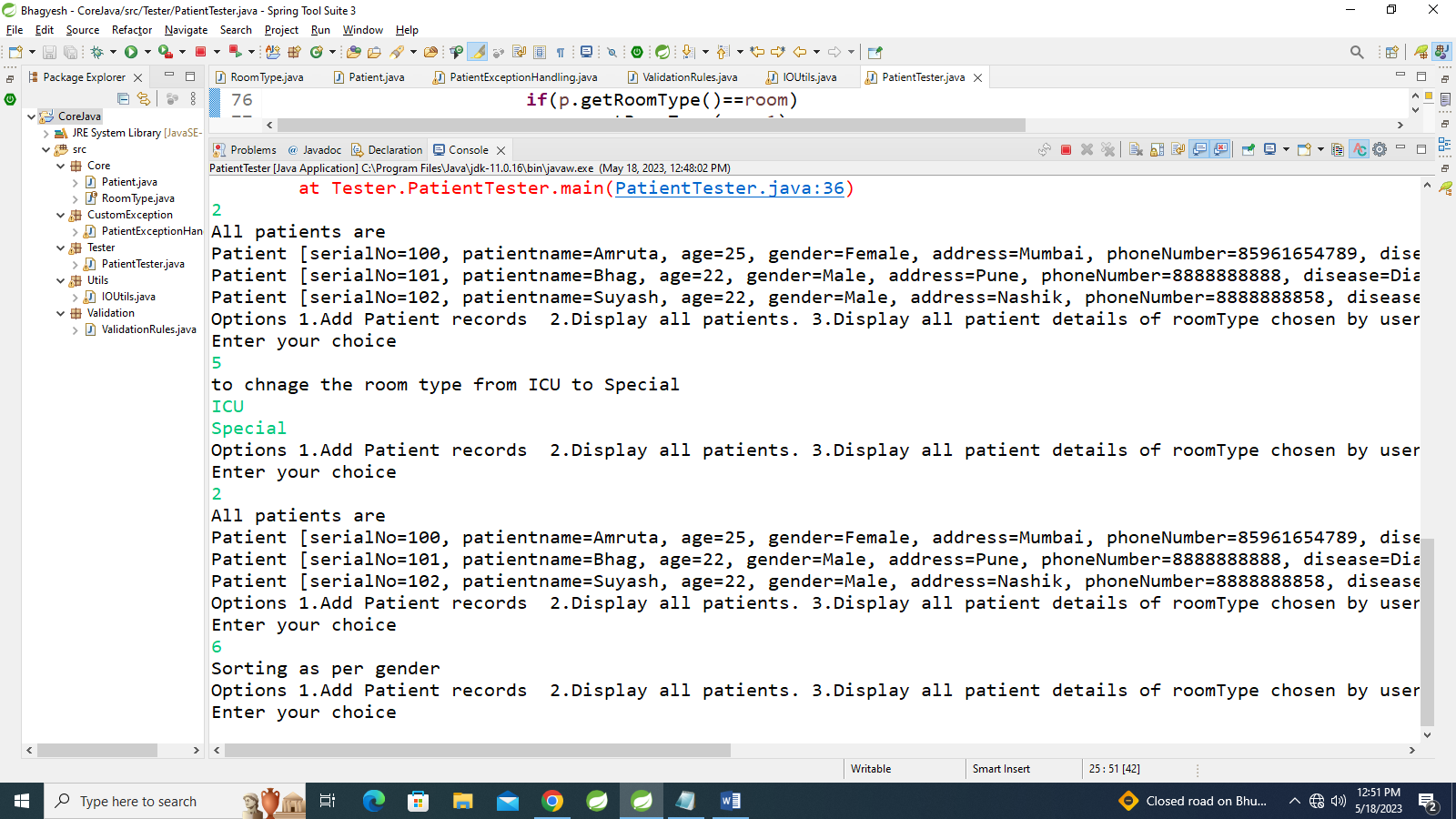


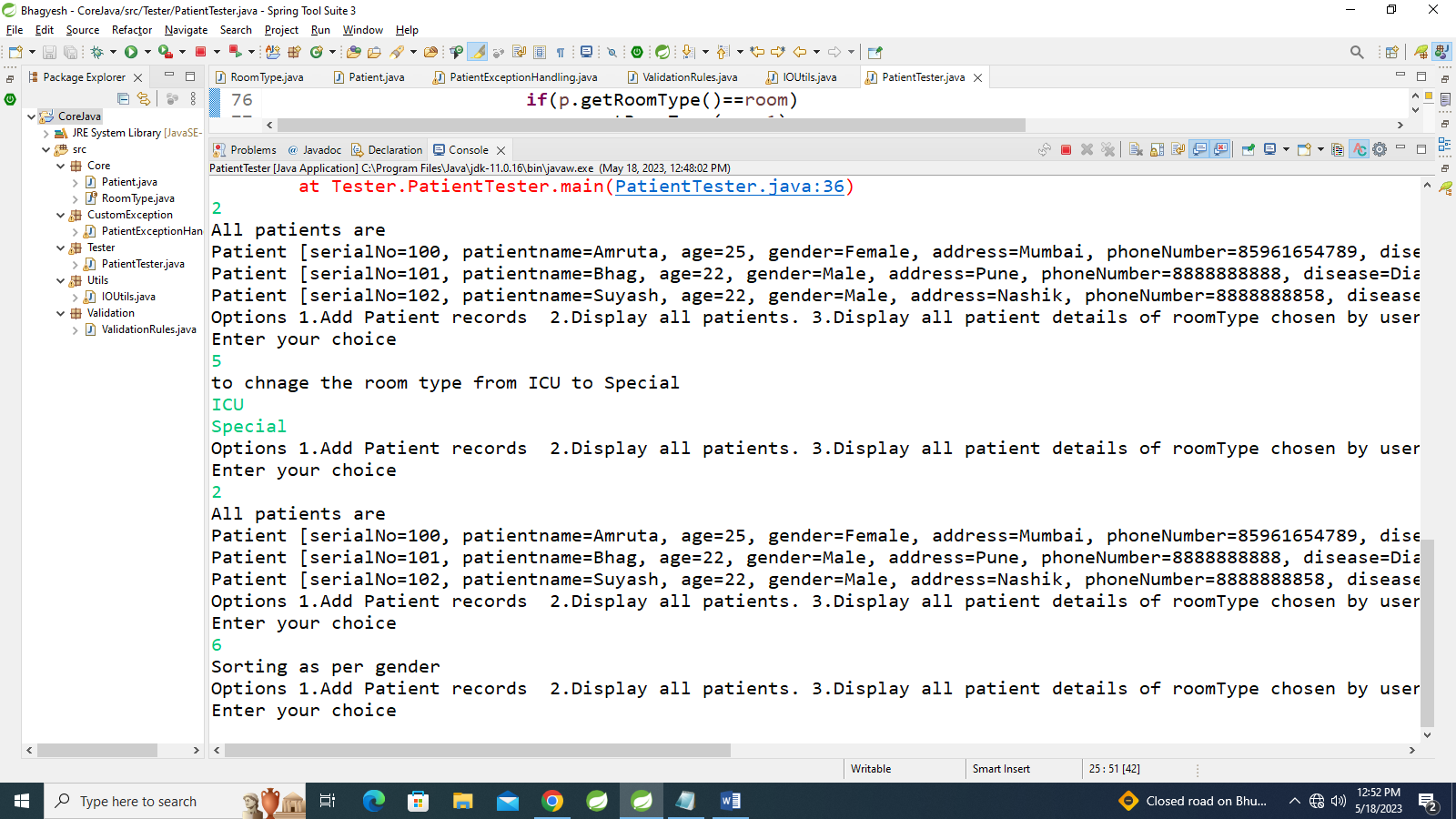
2.6)







\



**package** Tester;

**import** Core.Patient;

**import** java.time.LocalDate;

**import** java.util.ArrayList;

**import** java.util.Collections;

**import** java.util.Comparator;

**import** java.util.Iterator;

**import** java.util.List;

**import** java.util.Scanner;

**import** **static** Validation.ValidationRules.*validateAllInputs*;

**import** **static** Validation.ValidationRules.*parseAndValidateRoom*;

**import** Core.RoomType;

**import** **static** Utils.IOUtils.*storePatientDetails*;

**public** **class** PatientTester {

**private** **static** **final** String ***Diabetese*** = **null**;

**public** **static** **void** main(String[] args) {

**try**(Scanner sc=**new** Scanner(System.***in***))

{

System.***out***.println("Enter the fileName");

String file=sc.nextLine();

List<Patient>pat=**new** ArrayList<Patient>();

**boolean** exit=**false**;

**while**(!exit)

{

System.***out***.println("Options 1.Add Patient records 2.Display all patients. 3.Display all patient details of roomType chosen by user"

+ "4.Remove all patients who has disease Diabetese 5.Change roomType of patient who is from ICU to SpecialRoom"

+ "6.Sort patients based on gender 0.Exit ");

System.***out***.println("Enter your choice");

**try** {

**switch**(sc.nextInt())

{

**case** 1:

System.***out***.println("Enter patientname, age, gender, address, phoneNumber, disease,\r\n"

+ "admitDate(yyyy-mm-dd), annualIncome, roomType");

//String patientname, int age, String gender, String address, String phoneNumber, String disease,

//LocalDate admitDate, double annualIncome, RoomType roomType

Patient patient = *validateAllInputs*(sc.next(),sc.nextInt(),sc.next(),sc.next(),sc.next(),sc.next(),sc.next(),sc.nextDouble(),sc.next());

pat.add(patient);

**break**;

**case** 2:

System.***out***.println("All patients are");

**for**(Patient p: pat)

System.***out***.println(p);

**break**;

**case** 3:

System.***out***.println("Enter the room type");

RoomType roomType = *parseAndValidateRoom*(sc.next());

**for**(Patient p: pat)

**if**(p.getRoomType()==roomType)

System.***out***.println(p);

**break**;

**case** 4:

System.***out***.println("To remove all patients with diabetes");

Patient Dieseas1=(**new** Patient(sc.next()));

Iterator<Patient>itr=pat.iterator();

**while**(itr.hasNext())

**if**(itr.next().equals(Dieseas1))

itr.remove();

**break**;

**case** 5:

System.***out***.println("to chnage the room type from ICU to Special");

RoomType room = *parseAndValidateRoom*(sc.next());

RoomType room1 = *parseAndValidateRoom*(sc.next());

**for** (Patient p:pat)

**if**(p.getRoomType()==room)

p.setRoomType(room1);

**break**;

**case** 6:

System.***out***.println("Sorting as per gender");

Collections.*sort*(pat, **new** Comparator<Patient>() {

@Override

**public** **int** compare(Patient o1, Patient o2) {

**return** o1.getGender().compareTo(o2.getGender());

}

});

**break**;

**case** 0:

exit=**true**;

*storePatientDetails*(file,pat);

}

}**catch** (Exception e)

{

e.printStackTrace();

sc.nextLine();

}

}

}

}

}